Amendments to the Specification:

Page 1, amend first paragraph as follows:

B1

The present application claims priority of the Provisional Application ~~(RAL920000112US1)~~ filed on November 29, 2000, serial no. 60/253,869.

Page 5, amend paragraph beginning at line 2 as follows:

B2

Figure 1 shows a block diagram of a network processor in which ~~the invention according to the teaching of~~ the present invention is embedded. ~~Even though the~~ The present invention can be used in any technology in which data has to be stored and transmitted with sufficient BandWidth (BW) to support a FAT PIPE. As used herein a FAT Pipe is a high speed or high bandwidth data channel. It works well in the Network Processor environment and as such will be described in this environment. However, this should not be construed as a limitation on the scope of the present invention since it is well within the skill of one skilled in the art to make minor changes or no changes to adapt the teachings to other technologies.

Page 13, amend paragraph beginning at line 8 as follows:

B3

Still referring to Figure 3, the arbiter controller [[34]] 38 collects read/write requests from transmitter controller 30, receiver controller 26, and EPC controller 28 and schedules access towards individual memory store slices. The type of requests and structure of the requests from each of the named requesters are shown by arrows in Figure 3. As will be discussed in detail below, the data store memory is organized in units of 64 bytes (equal buffers). Frame data are then written into different buffers sprayed over different slices in order to maximize use of memory BandWidth (BW).

Page 13, amend paragraph beginning at line 15 as follows:

B4

In particular, receiver controller (Figure 3) receives data from incoming data interfaces and issues write requests in order to write receive data into individual buffers of the memory store. Similarly, transmitter controller 30 issues read requests in order to transmit selected frame on outgoing interface. Receiver and transmitter controllers are two functional blocks whose ~~functionally~~ functionality are determined by mode of operation of the data flow chip. These modes of operation can be line mode or switch mode. In line mode configuration receiver and transmitter controllers receive/transmit frames from/to line interface. In switch mode configuration, receiver and transmitter controllers receive/transmit PRIZMA cells from/to the switch interface. Typically, in Ingress direction receiver controller shall be configured in line mode and transmitter controller in switch mode. Similarly, in Egress direction receiver controller shall be configured in switch mode and transmit controller in line mode. In addition, line/line and switch/switch configurations can be supported.

Page 14, amend paragraph beginning at line 14 as follows:

B5

Referring again to Figures 2 and 3, the buffers for storing information are provided by DDR DRAM memories identified as slice 0 through 5. As will be described hereinafter, each memory slice has its own memory interface bus of 32 bits at 166 mHz. Frame data is written into the data buffer in chunks of 64 bits equal of buffer. Each buffer contains four Quad Words (QWs) corresponding to four banks of DRAM. Hence, one buffer equals 4 QW equals 4x16 bytes of data. Each buffer of the data store memory is addressed by buffer control block address (BCBA). To each BCBA and hence each buffer in the data store corresponds one BCB entry in the BCB memory (QDR SRAM) containing all information related to data in that particular buffer and chaining pointer for following buffer in the frame (or free BCB queue). The access window (read or write of B6 64 bytes) on DDR-DRAM interface consumes 11 cycles, which means that aggregate throughput of one slice is about 7.75 Gbps. In order to provide support of 10 Gbps on the FAT pipe port or 14.3 Gbps on switch interface, the arbiter controller, in the arbiter, accesses multiple slices of memory simultaneously. As a consequence, the writing algorithm has to spray (i.e. write) buffers belonging to one frame over all slices in order to prevent collision and starving of the FAT pipe port. Likewise, data is read simultaneously from multiple slices of memory ~~data~~.

Page 15, amend paragraph beginning at line 19 as follows:

B6 R/W requests from the EPC controller fills remaining opportunities within the access window to memory slice as well as complementing R and W access of transmitter and receiver at QW (Quad Word) level. It should be noted that a QW is 16 bytes of data.

Page 18, amend paragraph beginning at line 19 as follows:

B7 Still referring to Figure 3, the Transmitter Controller 30 reads frames from buffers in DDR DRAM and transmits them to an external bus interface which can be the Fat Pipe 22 for egress Dataflow 10B, or Switch Bus 20 [[21]] for ingress Dataflow 10A. The TP/TB QDR SRAM stores linked list data structures that maintain queues of frames awaiting transmission by the Transmitter Controller 30. When the EPC 12A/12B or Scheduler 14B/16A enqueues a frame for transmission, the EPC Interface Controller 28 writes the TP/TB QDR SRAM to enqueue the frame to the tail of the appropriate Target Port (TP) or Target Blade (TB) queue. The Transmitter Controller 30 performs a read of the TP/TB QDR SRAM to dequeue a frame from the head of a TP or TB queue. It then B7 reads the BCB Lists QDR SRAM to obtain the linked list of buffer addresses that form the frame. The Transmitter Controller 30 then reads the specified buffer addresses in DDR DRAM and transmits the frame data via Fat Pipe 22 or Switch Bus 20. As the data from each buffer is transmitted, the Transmitter Controller writes the BCB Lists QDR SRAM to place the buffer on the tail of the free queue buffer list.

Page 19, amend paragraph beginning at line 12 as follows:

Figure 4 shows a functional representation of the data flow chip and the data store that provides data to the FAT pipe port <u>31</u> coupled to the bus labelled data-out. For B8 brevity, elements in Figure 4 that are identical to previously named elements are identified by the same numeral. In light of this relationship receiver controller 26 (Figure 2) is identified by acronym SDM. Likewise, EPC interface controller 28 (Figure 2) is identified by acronym EPC and transmitter controller 30 (Figure 2) is identified by acronym PMM.

Page 20, amend paragraph beginning at line 12 as follows:

In one embodiment each of the buses 0' through N' are 32 bits wide and operate at the speed of 166 mHz. As a consequence, the throughput on each of the buses is approximately 7.75 Gbps. Each of the DDR DRAM such as slice 0, slice 1, slice 2 or B9 slice N are partitioned into a plurality of buffers. In Figure 4 one of the buffers in each slice is shown and are labelled buffer 0" 1" and 2". Each of the buffers span the four banks A through D that make up each slice of the data store. Each of the buffers 0"

B9

through N" are further divided into four smaller buffers labelled A' B' C' D' slice 0. Similar sub-partitions (not shown) are also used in the other slices of the data store. In one embodiment of the present invention the buffer 0", 1", etc. are 64 bytes whereas the smaller buffers A' B' C' D' are sometimes referred to as QWs are 16 bytes. As is evident in the figure, buffer [[0']] 0" is made up of four QWs, each being 16 bytes in size.

Page 20, amend paragraph beginning at line 20 as follows:

B10

Figure 5 is a graphical representation of the priority assigned to the SDM, EPC or PMM when accessing memory. The PMM is shown to have the highest priority 1, followed by SDM priority 2 and EPC priority 3. The priority are only exemplary and should not limit the ~~scrope~~ scope of the invention. The PMM reads buffers in any slice required. In Figure 5 the PMM is reading a buffer in slice 1. If required, to satisfy the FAT port bandwidth, the PMM could also access, in parallel, all or any of the other slices such as slice 1, slice 2 or slice 3 through N. Simultaneously, with the PPM reading slice 1, SDM is writing into slice 2 while EPC is reading from slice 0. Since each of the functional elements are accessing different slices in the memory, all of the named activities could occur in a single memory cycle.

Page 21, amend paragraph beginning at line 15 as follows:

B11

Still referring to Figure 5, when the PMM requires access to memory, it requires a bandwidth of approximately 10 Gbps. Any bandwidth less than the 10 Gbps (requirement of FAT pipe port) would cause corruption of data during transmission. To ensure the FAT Pipe port does not underrun a FIFO buffer 25 ~~(not shown)~~ with bandwidth greater than that of the FAT Pipe port, in one embodiment approximately 15.5 Gbps, feeds the FAT port. To maintain the 10 Gbps bandwidth buffers are accessed in the data store and

B11

stored in the FIFO 25 ~~(not shown)~~ at an average bandwidth of 10 Gbps. Each slice has a bandwidth of 32 bits. Each slice interfaces into a 64-bit bus (32-bit DDR physical interface to each memory, 64-bit internal bus inside the network processor after demultiplexing of the DDR bus).